# Review on Containerization Technology for Edge-Cloud Applications

Amritha Soloman
Mtech CSE Dept.
Mangalam College of Engineering
Ettumanoor, Kottayam, India

Dr. John K. Peter
Prof.,CSE Dept.
Mangalam College of Engineering
Ettumanoor, Kottayam, Imdia

***Abstract*— Recently, containers have shown to be an effective lightweight technology for virtualizing applications, especially for managing apps in the cloud. A common issue is the orchestration of the building and deployment, which frequently necessitates the administration of container clusters. Researchers have focused on this new subject, however there isn't a secondary research to support this work at the moment.**

**A prevalent practise in the IT industry nowadays is the usage of containerization systems like Docker, which are regularly combined with virtual machines in cloud-based situations. Embedded control and monitoring device containerization platforms have been used significantly less frequently so far. For embedded control systems, however, the benefits of greater modularization and mobility are relevant. Our goal is to contrast the body of existing research on container orchestration, specifically how this technology is used in the cloud. In this article, we present a performance evaluation approach and use it with edge-cloud computing apps that use containerization**

***Keywords—Virtual Machine, Docker, Performance evaluation, Containerization, Industrial Internet of Things.***

## I. INTRODUCTION

Over the past ten years, application containers have gained popularity in the IT industry. A container image is a single, standalone package that houses all the software binaries and information about all the dependencies of a containerized application. In virtualization scenarios, when software is segregated from the platform it runs on, containers are employed. The development of a virtual replica of a device or resource, such as a storage device or a network resource, is referred to as virtualization. Additionally, the virtual component is frequently segregated from the host system, so if it fails, the host system is unaffected. Applications can be enclosed into containers that can be readily deployed on numerous different platforms if a containerization platform is chosen and containers cause no compatibility issues with required dependencies. After many years of rapid expansion, the traditional centralised cloud-enabled Internet of Things (IoT) architecture has faced enormous challenges, such as the heterogeneity of hardware, software, communication protocols, and data formats, and especially the demanding low latency and high reliability needed by time-critical applications in industrial IoT. (IIoT). As a consequence of the technical advancement in cloud computing, the edge-cloud computing paradigm has been presented recently to address the issues and make it easier for IoT innovations to be adopted into multiple domains, as shown in fig 1.

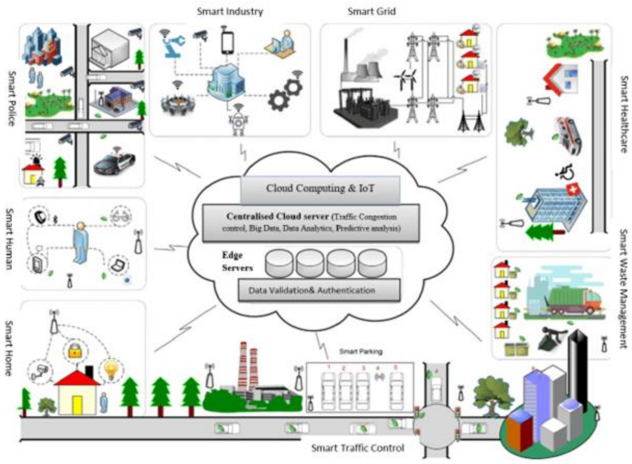


Fig: 1 Fig: 1 a description of the edge-cloud computing paradigm and how it is applied in many contexts, including smart cities, agriculture, logistics, transportation, factories, power grids, and healthcare. While field and edge devices communicate via a variety of wired or wireless protocols, edge devices are connected to the cloud platform through Ethernet.

The edge-cloud computing paradigm, which aims to allow quick decision-making and intelligence at the network edge, is starting to acquire more and more traction today, especially for industrial applications. The implementation of edge-cloud computing infrastructures has been put on the industry's roadmap for the container-based virtualization technology.

In this research, we present a performance assessment technique from the viewpoint of the industrial customer and apply it to the cutting-edge edge-cloud computing infrastructures based on containerization.

According to the findings, edge containerization does not significantly reduce communication, computation, or intelligence capabilities, making it a potential solution for the edge-cloud computing paradigm. However, there is still a significant performance gap between the edge-cloud architecture as it is now implemented and the rigorous specifications required by time-sensitive industrial applications.

We also stress and demonstrate how dividing an industrial application into microservices throughout the whole stack may be taken into account when designing a solution. Users of edge-cloud computing and developers may both utilise the suggested assessment approach as a guide to gain a client-side viewpoint overview of system performance. Process automation, industrial control, power system automation, building automation, and automobiles, among other diverse industrial sectors and IIoT end applications, have all contributed to the rich innovation of industrial networking protocols. As defined by the open systems interconnection (OSI) model, the connectivity between industrial field devices and edge infrastructure varies depending on the communication medium and protocols from the media access control (MAC) layer all the way up to the application layer, resulting in significantly different performance. Numerous academics have conducted in-depth analyses of the effectiveness of current industrial connection protocols from various angles, including IEEE 802.11, IEEE 802.15.4, and ISA100.11a, but the effectiveness of the backbone, or the edge-cloud computing stack, is still unclear.

The main contributions of this study are as follows.

Carried out an evaluation from the viewpoint of the client of the holistic system latency performance of containerization, namely full stack round trip time with regard to message sending interval, payload, network bandwidth, and concurrent devices, on the cutting-edge edge-cloud computing platforms.Evaluated how well edge-cloud computing stacks' containerization could process machine learning and neural network activities for actual industrial applications.

Highlighted the performance constraints and emphasised the need for partitioning in the edge-cloud computing infrastructures now in use in industrial practise..

## II. RELATED WORKS

By utilising one or more of the devicecloud, device-edge-cloud, and device-edge deployment models, numerous studies have been carried out to examine the performance of containerization in the edge-cloud computing stacks from the networking layer, application layer, or resource utilisation perspective. For instance, several studies have looked into how well the container technology performs in terms of networking.

Claus Pahl and colleagues conducted a systematic mapping investigation of 46 chosen studies as part of their 2017 study on cloud container technologies. On the basis of a characterization framework, he categorised and contrasted the chosen research. By situating container orchestration inside the framework of the cloud while simultaneously bringing it closer to current problems in cloud platforms, microservices, and continuous development, this leads to a discussion of shared and growing challenges in the container orchestration area.

An assortment of technology assessments, solutions, and use case architectures make up the scientific contributions that were examined (conceptual and implemented). This might be seen as a symptom of the immaturity of a developing area because a large portion of it is still conceptual. The lack of extensive empirical assessments and the limited number of use case validations support this conclusion.

Additionally, there is a glaring disparity in contribution forms as compared to more established domains:

• More technical contributions (solution suggestions): There are fewer journal articles (with short communications in magazines published only).

• More use cases than technical solutions: Using use cases rather than summative evaluations to formatively evaluate new technologies

It has been established that containers are more flexible as an application management framework and are more resource efficient than virtual machines (VMs). To properly enable a container design for cloud settings, however, there are not enough tested technologies available. One of the examples given is failure management. Better resource monitoring and log analysis are necessary for identifying underlying causes and handling unusual situations. Conclusion: While failure management is still lacking, the area is developing toward container middleware (and even container PaaS), with isolation, construction, quality management, orchestration, and distribution management as its primary priorities.

2019 saw research on Emerging Trends, Techniques, and Open Issues in Containerization from IEEE Senior Member Junzo Watada. The knowledge gap between developers and the available technologies is further widened by the fact that a thorough review of containerizations and their popularity dynamics has yet to be discovered in modern literature. Therefore, the current study examines numerous technical aspects of containerization along with potential issues and prospective solutions as well as various significant industrial applications to demonstrate its current supports and technological challenges.

Finally, we have conducted a thorough experimental study to compare the performance of virtual machines, containers, and unicernel using standard benchmarks and observed containers to deliver satisfactory performance in almost all aspects, but are still not free from issues regarding isolation & security, performance stability, lack of available effective tools for c. Unikernels offer VM-like isolation and high performance, but they still need to develop the necessary technological sophistication (in terms of microprocessor stability, process containment, persistent storage, etc.). On the other hand, VMs are shown to offer consistent performance throughout, however their primary drawbacks continue to be their larger memory footprints and slower spin up/down.

Performance Evaluation of Containerization Platforms for Control and Monitoring Devices was the subject of research conducted by Harri Manninen et al. in 2020. a Performance Evaluation of Containerization Platforms for Control and Monitoring Devices was developed. initial assessment findings for several containerization platforms for embedded control and monitoring systems were reported. Docker, Balena, LXC, and Flatpak were all examined. Memory use, CPU usage, and power consumption have been the main topics of our evaluation. As can be observed, containerization generally does not increase an application's CPU, memory, or power consumption significantly. In terms of memory use, Docker appears to be the platform that consumes the most resources, Because of this, they think that containerization may be prepared for embedded software that runs on systems with properties akin to those of the Raspberry Pi 4. We anticipate that because his apps are I/O-intensive and relatively small, there will be even less overhead for bigger applications. They also limited applications to just three. More containers would generally be present in conventional IT scenarios, but we think this lower number more accurately represents an embedded systems environment.

A performance evaluation of containerization in edge-cloud computing stacks for industrial applications was carried out in 2021 by Yu Liu et al. In this work, they present a performance evaluation approach from the viewpoint of an industrial customer and use it with cutting-edge edge-cloud computing infrastructures based on containerization. On entire stack latency, the effects of message sending interval, payload, network bandwidth, and concurrent devices are examined, and the computing power required to carry out machine learning tasks is benchmarked. According to the findings, edge containerization does not significantly reduce communication, computation, or intelligence capabilities, making it a potential solution for the edge-cloud computing paradigm. However, there is still a significant performance gap between the edge-cloud architecture as it is now implemented and the rigorous specifications required by time-sensitive industrial applications. Additionally, underline and demonstrate how partitioning an industrial application into micro services throughout the whole stack may be taken into account when designing a solution. Users of edge-cloud computing and developers may both utilise the suggested assessment approach as a guide to gain a client-side viewpoint overview of system performance.

## III. EDGE CLOUD ARCHITECTURE

An IT environment known as a cloud pools, abstracts, and distributes IT resources over a network. A network's edge computing location, along with the associated hardware and software, is referred to as an edge. Running workloads within clouds is known as cloud computing, whilst running workloads on edge devices is known as edge computing.

If the storage and processing resources offered by edge devices at network endpoints are abstracted, pooled, and shared across a network, they can contribute to a cloud by basically becoming a part of a larger cloud architecture.

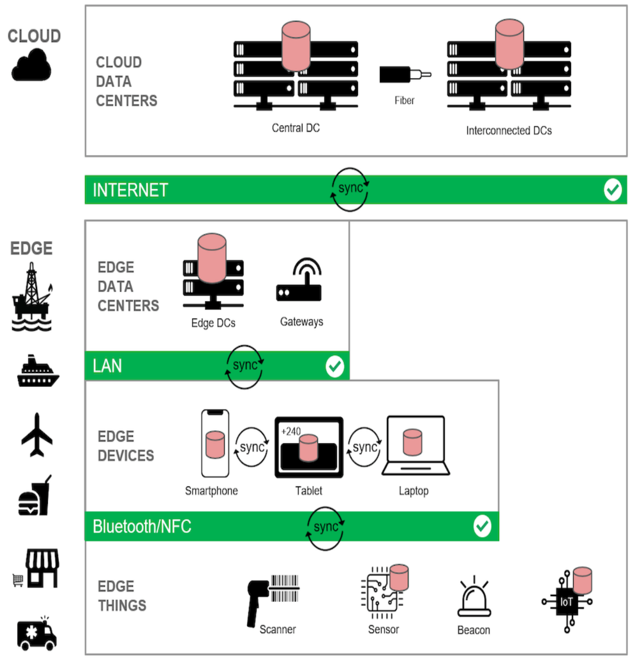


Figure 2: Edge Cloud Architecture

A cloud does not include edge computing. Edge computing's deliberate separation from clouds and cloud computing *is what makes it so beneficial.* Applications may operate and data can be stored in clouds. They are server farms or datacenters-created software-defined environments.

Additionally, data is gathered at edges. They are actual physical spaces built with hardware outside of a datacenter.

- Running workloads in a cloud is the act of cloud computing.
- Running workloads on edge devices is a form of edge computing.

Edge computing is not the same as an edge (location) (action). Edge computing is not the process of gathering data at the edge of a network and sending it, with little to no alteration, to the cloud. This is merely networking. Edge computing, however, refers to the collection and processing of data at the edge.

For the following two key reasons, edge computing differs from clouds:

Sensitivity to time. Decisions must be taken quickly, thus there is no time for the latency that would typically occur as data is gathered by an edge device, sent to a central cloud without alteration, processed, and then delivered back to the edge device for execution.

Data amount. Too much data has been gathered to transfer it unmodified to the cloud. They can all be related. They don't have to be connected, though.

Clouds can survive without edge devices or the Internet of Things (IoT). Edge and IoT can function without clouds. IoT is not dependent on edge computing or edge devices. IoT devices could link up with a cloud or an edge. Some edge devices have a constant connection to a cloud or private datacenter, while others have sporadic connections to similarly central places or have no connections at all.

However, edge computing seldom exists without IoT whether it is employed in manufacturing, mining, processing, or shipping processes. This is so that edge devices may process and activate sources and destinations without relying on a centralised location or cloud. IoT devices are commonplace physical things that gather and transmit data or command activities like manipulating switches, locks, motors, or robots.

For Example: IoT activities often include home automation. Because they simply transmit data and execution decisions back and forth, our phone and smart home appliances (lightbulbs, thermostats, and outlets) are all Internet of Things (IoT) devices (sometimes through a cloud). Your phone and other smart gadgets do not process the data they gather.

An edge computing exercise involves satellite imagery, such as that utilised on the International Space Station (ISS). The IBM Cloud on Earth is connected to edge devices that are physically located on the ISS and executing containerized analytical programmes as a single-node Red Hat OpenShift cluster. Only photographs worth sending down to the ground are sent. The sheer amount of data collected is too much for an Earth-based cloud to handle, hence edge computing is a step that is required in this situation.

Compute resources and services are frequently consolidated at sizable datacenters in a cloud computing approach. A piece of the network infrastructure needed to link IoT devices to the internet is frequently provided by clouds. To enable remote administration, to accept automation commands, to pass network telemetry traffic required for analytics, and to deliver data information which will be recently saved in databases, and evaluated to achieve corporate objectives, edge devices need network access to central locations.

An edge device could send a log of the decisions it made back to the datacenter for data storage, data management, data processing, or big data analysis as part of the communication offered by a cloud service. Alternatively, the communication could simply be the transfer of data from an edge device across a cloud and into a datacenter.

Industrial IoT, often known as IIoT, is the application of IoT in an industrial setting, such as manufacturing machinery. Consider the lifespan of the large, factory-used machinery. Equipment may be stressed differently over time depending on the user, and malfunctions are a normal aspect of operations.

The sections of the machinery that are most prone to damage or misuse can be equipped with IoT sensors. Predictive maintenance may be performed using the data from these sensors, cutting down on overall downtime.

The necessity for edge computing and IoT technologies to collaborate is demonstrated by autonomous cars. In addition to keeping an eye on the car's systems, an autonomous vehicle travelling along the road has to gather and interpret real-time data regarding traffic, pedestrians, street signs, and stop lights.

Sending data back and forth from the car to the cloud for processing would be too time-consuming if the vehicle needed to stop or turn rapidly to prevent a collision.

The IoT sensors in the car can analyse the data locally in real-time to prevent an accident thanks to edge computing, which delivers cloud computing capabilities to the vehicle.

## IV. EDGE CONTAINERS

To minimise latency, conserve bandwidth, and improve the overall digital experience, edge containers are decentralised computing resources placed as near to the end user as feasible.

Every day, there are more and more gadgets with Internet connection. As a result of the IoT revolution, we now have smart TVs, smart houses, smart phones, smart automobiles, and many more smart objects. Additionally, since 2015, the majority of internet traffic has come from mobile devices.

The majority of consumers utilise time-sensitive applications, therefore latency degrades the user experience. High latency remote centralised cloud services are frequently to blame for subpar application performance. Edge computing was created to alleviate network performance difficulties and move data processing closer to the consumer.

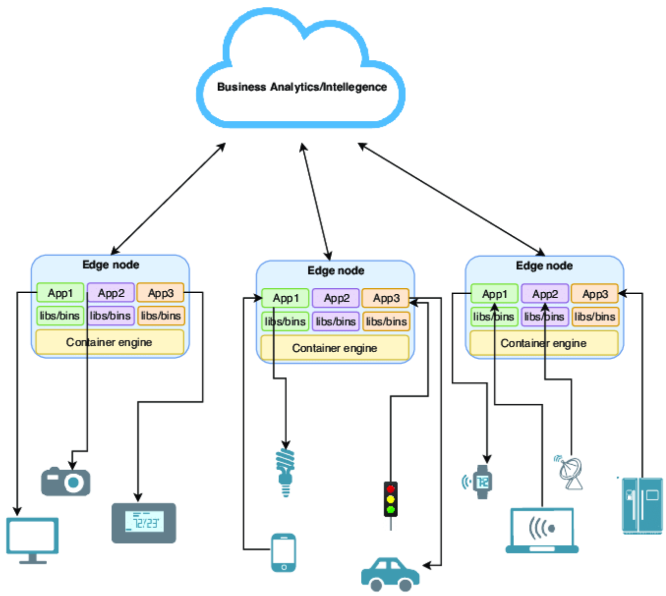


Figure: 3 Container based Edge Computing

By shifting important parts of their application to the edge of the network, edge containers in particular enable enterprises to decentralise services. Businesses may cut their network expenses and improve reaction times by moving intelligence to the edge.

Containers are a natural match for edge computing solutions since they are simple to install as software packages and containerized apps are simple to distribute. When compared to a typical cloud container, edge containers can be deployed in parallel to geographically different points of presence (PoPs) to achieve greater levels of availability.

Location is the primary distinction between edge containers and cloud containers. Edge containers are situated at the network's edge, significantly closer to the end user than cloud containers, which operate in distant continental or regional data centres.

Edge containers employ the same technologies as cloud containers because the location is the key distinction, allowing developers to use their current Docker experience for edge computing. Organizations can utilise a Web UI, Terraform, or a management API to manage containers.

Probes and real-time metrics may be used to track and evaluate the utilisation of edge containers.

Edge containers provide advantages.

Low latency: Because edge containers are situated within a few hops from the end user, they offer extraordinarily low latency.

Global load balancing: Using a single Anycast IP, traffic may be globally dispersed to the nearest container.

Scalability: Compared to a centralised cloud, an edge network contains a large number of PoPs. The ability to simultaneously deploy edge containers to several locations allows businesses to better fulfil local demand.

Maturity: Technologies for containers, like Docker, are regarded as mature and tried-and-true. Developers testing edge containers may use the same Docker tools they are accustomed to using because no new training is needed.

Reduced bandwidth: Because all traffic is concentrated in the data centre of the cloud vendor, centralised apps may have high network costs. Edge containers can offer pre-processing and caching since they are situated near to the user.

Edge containers vs other edge computing solutions:

Comparing serverless functions to Edge Serverless, developers are restricted to the runtimes and languages offered by vendors. Serverless functions are easier to deploy. On the other side, containers may execute any unique programme in any language. A pay-as-you-go paradigm applies to serverless, but a set monthly fee is more common for containers.

Compared to Edge VMs, there is a set monthly fee depending on machine size for both containers and VMs. While VMs provide developers a fully functional machine that increases flexibility, they also come with the added responsibility of OS administration. Additionally, unlike the underlying systems in containers are controlled by the manufacturer, VMs need patching and system administration. Examples of Edge Containers

Edge computing has the potential to provide businesses an advantage over their rivals. A game-changing startup called Edgegap introduced a technology in 2019 that employed real-time telemetry and edge containers to cut latency by 58%.

The proprietary Edgegap technology dynamically locates the instance of multiplayer games, reduces latency, and enhances the player experience by using the power of numerous locations deployed in an edge computing environment.

Edge containers not only serve the multiplayer gaming space. Other uses cases for edge containers include:

- Internet of Things (IoT) devices
- Real-time video and voice recognition
- Real-time analytics
- Real-time processing of sensor and telemetry data
- Augmented reality
- Video and audio streaming

## V. PROPOSED SYSTEM

In this section, the architecture and typical deployment models of the Edge cloud computing for containerization based IoT applications are introduced.

Industry applications are modularized and handled as separate containers by a container runtime at the edge unit. With the help of container orchestration systems like Kubernetes, these containerized apps may be remotely generated, updated, and destroyed in an elastic way with little overhead. Similar to the function an IoT hub performs in the cloud, an edge hub infrastructure is responsible for maintaining traffic flows occurring at the edge, including message routing among containerized applications, bidirectional communication between field devices and the cloud platform, and field device-to-cloud platform communication. By establishing connections with the edge hub, either directly via natively supported protocols like the message queuing telemetry transport (MQTT) protocol or via a protocol translator, field IoT devices may be served by edge apps. The industry has adopted this design as the de facto edge-cloud computing architecture; examples include Microsoft Azure IoT Edge, Amazon Greengrass, IBM Edge Application Manager, and Huawei KubeEdge. Industry artificial intelligence and IoT (AIoT) applications are expected to take advantage of this architecture.

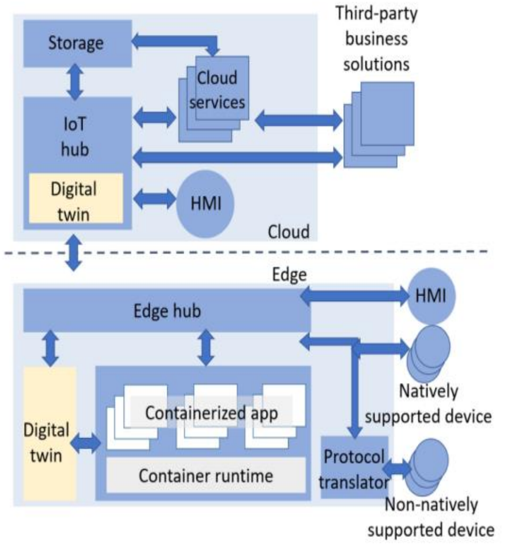


Figure 4: Architecture of Edge cloud computing for containerization based IoT applications

Containers are hence appealing for non-enterprise usage like IIoT edge devices. They may be updated independently on the same server without impacting other containerized apps since these containerized packages and their contents are partitioned from one another and the rest of the system.

## VI. RESULT ANALYSIS

Approaches to containerization offer benefits for system management and operation across physical compute resources. Containers are used in the business IT field to separate computational workloads from the computer platform on which they execute. By spreading out numerous workloads across racks and scaling the hardware resources, such as processors, memory, and storage, as required to support the workloads, for instance, computer hardware may now be thought of more as a utility.

It is possible to make better use of the hardware investment and increase system resiliency by multiplexing software loads over fixed hardware resources. By permitting plans where a centralised container provisioning or configuration can be altered and then sent out to the execution environment, it also makes it easier to maintain and evolve the software workloads themselves. The modern cloud has been significantly facilitated by containerization technologies as they have been applied to traditional business IT.

Some systems are more capable and offer more functionality when it comes to building and maintaining containers. A container may always be built "by hand," but there are frequently resources and tools available in an open source ecosystem that can help. A container may often be derived from a composition or library of reference containers in a contemporary system. These libraries encourage reuse, take use of the ecosystem, and enable quick container development and deployment.

In general, containerization techniques disentangle the difficulty of carefully controlling the provisioning of an application's execution environment from the underlying hardware computation resources. Benefits of partitioning, security, and orchestration are provided by containers. Although the method is less expensive than complete virtual machines, it may nevertheless duplicate operating system and userspace components and result in additional disc space, memory, and, to a lesser extent, CPU overhead. Containerd, Docker, and Snaps are a few well-known containerization technologies used in business IT. The well-known orchestration platforms are Kubernetes and Docker Swarm.

## VII. CONCLUSION

As a technology, containers enable the administration of Edge cloud IoT sensors and devices in a way that makes it simple to scale them up and down. Additionally, operating large hypervisors is not necessary with containers. By doing this, we can lighten the burden on IoT systems and increase their dependability.

There are countless possibilities for creating intelligent IIoT systems that span the edge and cloud. Therefore, before opting to embrace this technology, firms should be well-informed of the dangers and advantages. Do take into account a solution's container-nativeness while assessing an IoT solution. When managing IoT data, installing and maintaining applications and operating systems, and managing IoT data, containers have several advantages. IT and DevOps have undergone a revolution because to containers. Although it is still early, containers are poised to have a similar impact on the IoT industry.

In order to evaluate the effectiveness of the edge-cloud system and containerization technology on edge cloud applications, we thoroughly assessed the well-known containerized edge-cloud architecture.

## REFERENCES

[1] David Bernstein. Containers and cloud: From lxc to docker to kuber netes. IEEE Cloud Computing, 1(3):81–84, 2014J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Minxian Xu, Wenhong Tian, and Rajkumar Buyya. A survey on load balancing algorithms for virtual machines placement in cloud computing. Concurrency and Computation: Practice and Experience, 29(12):e4123, 2017.

[3] Cloud Container Technologies: a State-of-the-Art Review Claus Pahl∗ and Antonio Brogi: DOI 10.1109/TCC.2017.2702586, IEEE Transactions on Cloud Computing

[4] Performance Evaluation of Containerization in Edge-Cloud Computing Stacks for Industrial Applications: A Client Perspective YU LIU 1, DAPENG LAN 2, ZHIBO PANG Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/ VOLUME 2, 2021

[5] Performance Evaluation of Containerization Platforms for Control and Monitoring Devices Harri Manninen Aalto University Espoo, Finland harri.m.manninen@aalto.fi Vesa Ja¨askel ¨ ainen ¨ Vaisala Oyj Vantaa, Finland vesa.jaaskelainen@vaisala.com, 978-1-7281-8956-7/20/$31.00 ©2020 IEEE

[6] Yocto project. yoctoproject.org. https://www.yoctoproject.org/ (accessed Feb. 26, 2020).

[7] A. Furtado, J. Pacheco, and R. Oliveira, “PHY/MAC uplink perfor□mance of lora class a networks,” IEEE Internet Things J., vol. 7, no. 7, pp. 6528–6538, Jul. 2020.